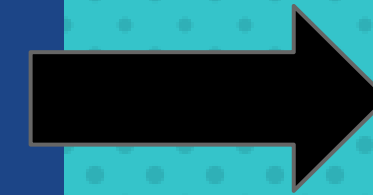


ORIENTAÇÃO A OBJETOS - MÉTODOS

Programação Orientada a Objetos



Abstração

Encapsulamento

Métodos

Construtores



Objetos

Atributos

Classes

Modificadores
de acesso

Métodos de
acesso

Métodos
modificadores

Getters e
Setters

Métodos
estáticos

Herança

Polimorfismo

Reutilização

Sobrescrita de
métodos

Associação

Extends

Super e
subclasse

Classes
abstratas

instanceof e as
operator

métodos
abstratos

interfaces

Generics

END

Teste de
Software

***NO PROJETO DA
AULA ANTERIOR...***

Algumas ações foram desenvolvidas para a nossa classe/simulador.

ADICIONANDO MÉTODOS

Vamos colocá-las em métodos, e considerando ainda o tempo de duração das ações, quando pertinente, para impactar os atributos.

***RECAPITULANDO UM
CONCEITO DA AULA
ANTERIOR...***



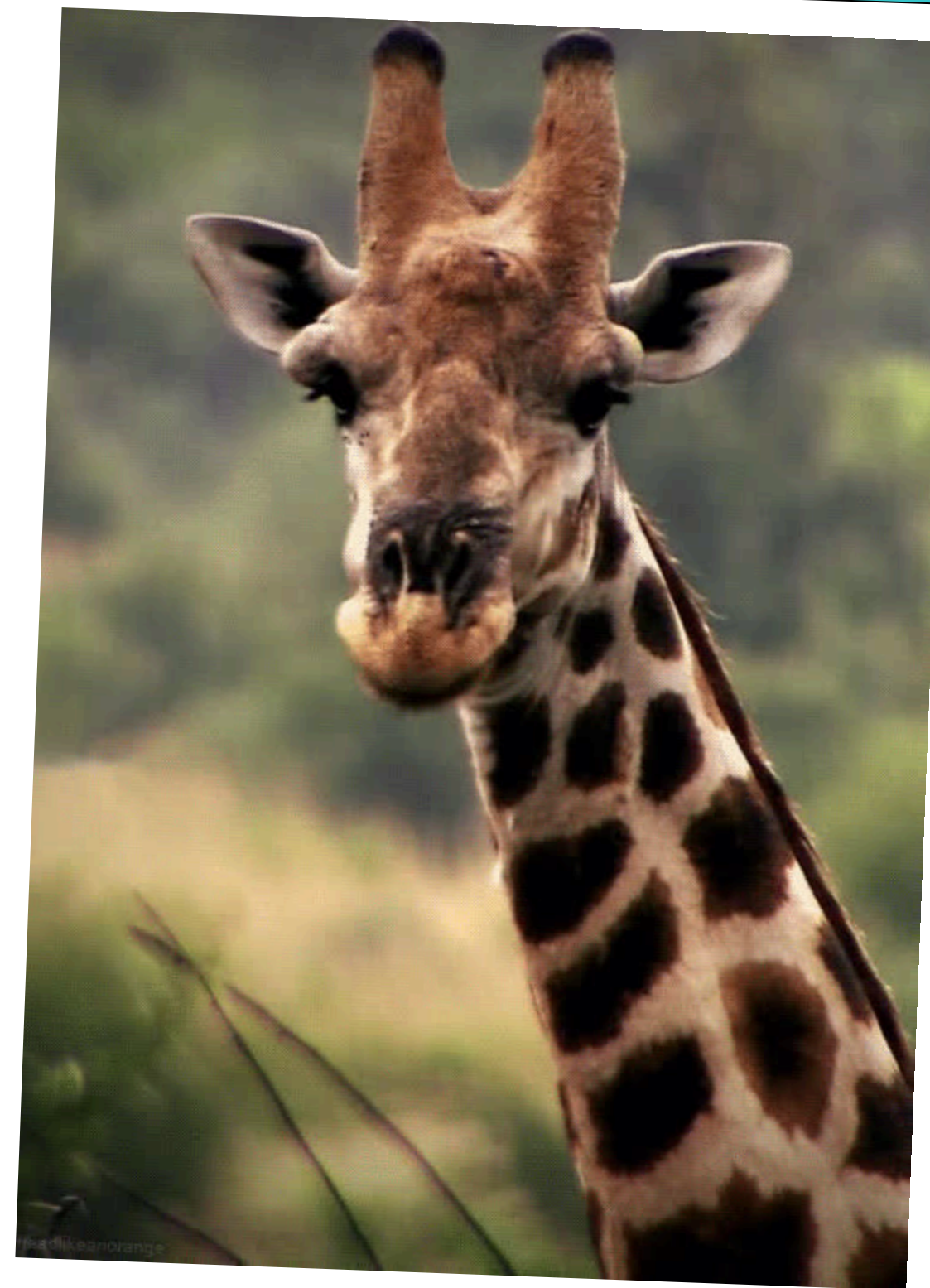
ORIENTAÇÃO A OBJETOS

Em uma boa orientação a objetos, as responsabilidades e validações que dizem respeito a uma classe X, estão na classe X.

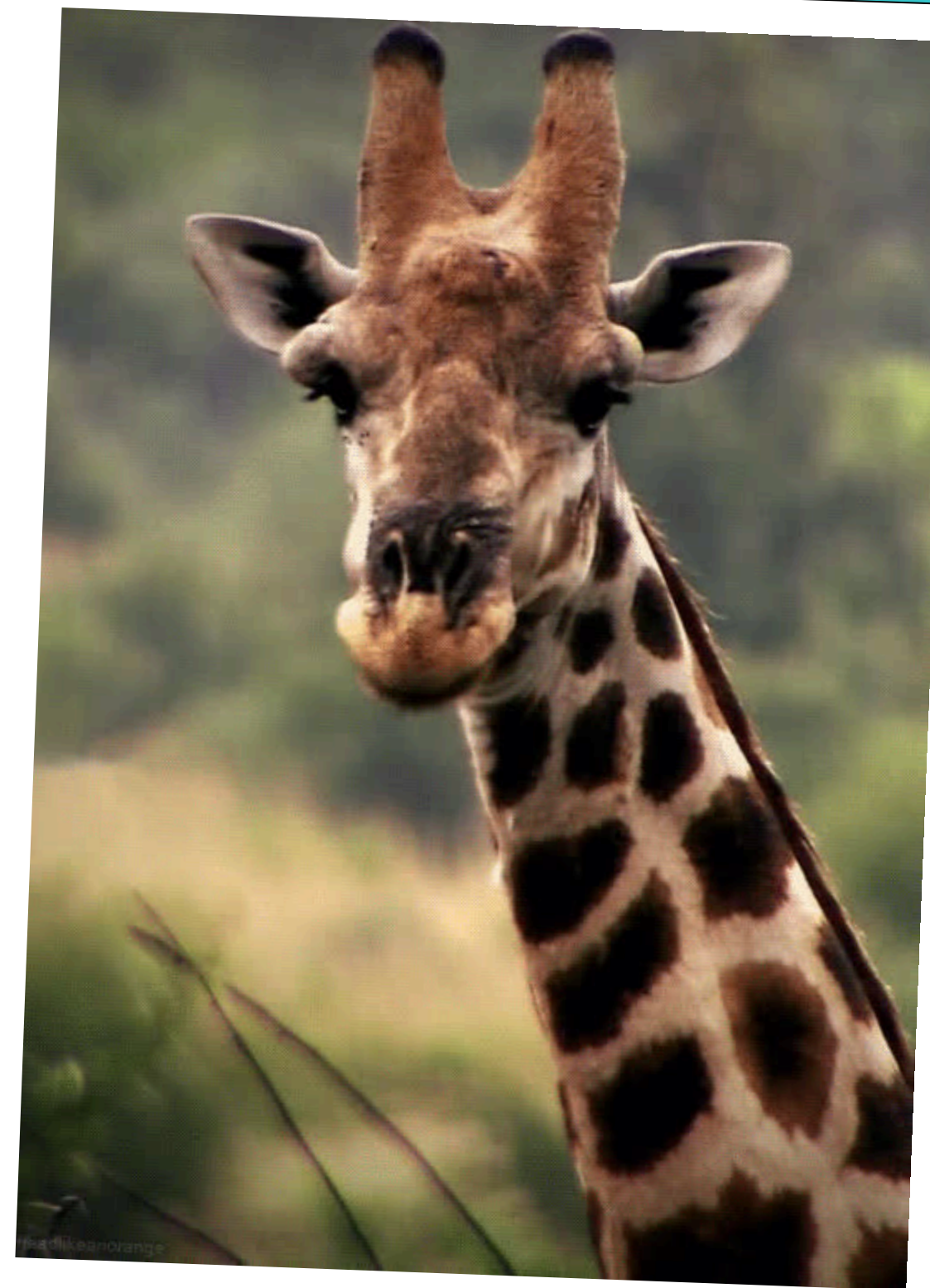
***LER E APRESENTAR DADOS SÃO
PAPÉIS DAS NOSSAS
CLASSES?***



**QUANDO FAÇO A LEITURA E SAÍDA NA
PRÓPRIA CLASSE ESTOU **ENGESSANDO**
QUE MINHA CLASSE SÓ PODERÁ SER
UTILIZADA NO TERMINAL**



MAS COMO RESOLVER ESSA TRETA?



MÉTODOS COM PARÂMETROS



PARÂMETRO

Os parâmetros especificam dados que são **necessários** para o funcionamento do método. Esses parâmetros podem ser de qualquer tipo de dado, seja ele primitivo ou referência.

**SENDO ASSIM, PODEMOS DIZER
QUE OS PARÂMETROS RESOLVEM
O PROBLEMA DA ENTRADA DE
DADOS DENTRO DOS MÉTODOS**

MÉTODO COM RETORNO



RETORNO DO MÉTODO

O retorno do método devolve a quem o invocou um dado que é resultado do processamento do mesmo

Este dado pode ser o resultado de um processamento ou uma mensagem a ser impressa, quando necessário

RETORNO DO MÉTODO

Quando é especificado que o método possui um tipo de retorno que não seja void, obrigatoriamente ele deve retornar um valor.

**SENDO ASSIM, PODEMOS DIZER
QUE O RETORNO RESOLVE O
PROBLEMA DA SAÍDA DE DADOS
DENTRO DOS MÉTODOS. AGORA
OS DADOS SÃO RETORNADOS**

RETORNO DO MÉTODO

Assim como os parâmetros, o retorno pode ser de qualquer tipo de dado. Pode ser um primitivo ou referência. Ou seja, o retorno pode ser um number, Funcionario, Cliente, Veiculo, string, etc.

Atividade – Criar e ser criativo!
As classes são apenas um guia de exemplo

Emprego

Função
Salario por hora

Pessoa

Nome
CPF
Data de Nascimento
Veiculo[]
Emprego
Carteira (dinheiro)
Horas trabalhadas

CriarPessoa()
comprarVeiculo(Concessionária,...)
venderVeiculo()
receberHorasTrabalhadas()

Concessionária

Nome
CNPJ
Veiculos[]

VerificarVeículos Disponíveis()
CadastrarVeículos()

Veiculo

Modelo
Marca
Ano
placa
Potencia
estado(lig/des)
Num. de marchas
Marcha atual
Tam. do tanque
Nível do combustível
Disponível
valor

Ligar()
Acelerar()
Parar()
aumentarMarcha()
diminuirMarcha()
Buzinar()
verificarDados()